ロング・フォーマットとワイド・フォーマットを駆使してデータ前処理を行う

―TECMIN が出力する CSV ファイルの変換を素材に―

To do the pre-process data analysis by using long format and wide format

―By converting the CSV file which is outputed by TECMIN―

藤　本　一　男（作新学院大学名誉教授）
永　島　　淳（作新学院大学 EMIR 室）
Fujimoto Kazuo（Sakushin Gakuin University, Professor Emeritus）
Nagashima Jun（Sakushin Gakuin University, EMIR office）

概要

本稿は、作新学院大学の学内情報サービス（TECMIN）が出力する CSV ファイルの形式に注目することで、その形式（ロング・フォーマット）が、このシステムに特異なものではなく、データ分析にとって汎用性を有する非常に重要なものであることを明らかにした。

その過程で、コマンドのパイプ処理による処理の連鎖、という流れの中で、このロング・フォーマット－ワイド・フォーマットの変換を伴う過程が、データの「前処理」過程の基本形であることを述べ、最後に、以下のようにまとめた。1）ワイド・フォーマット（いわゆる表計算型）は、最終形態としては意味をもつ。2）分析過程では、適用する統計処理に必要な形態に展開すべきである。3）データのマージ（連結）は、ワイド・フォーマットに執着することなく、ロング・フォーマットとの相互変換を活用して行う。4）データの前処理を行うための基本技法としてロング⇄ワイドの表変換をマスターすべきである。

キーワード：TECMIN、NetCommons、ロング・フォーマット、ワイド・フォーマット、tidyverse、EDA（探索的データ解析）

1　はじめに：問題の所在

1.1　TECMIN が出力する CSV ファイルのフォーマット

作新学院大学大学の学内情報サービス（TECMIN）[1] の本体は、NetCommons というサーバーで、開発は国立情報学研究所（Nii）である。2012年より利用されてきており、10年近い稼働実績をもっている。NetCommons の詳細は、URL：https://www.netcommons.org/ を参照。

その機能の１つに、アンケート機能もあり、学生の授業への反応を収集するのに、一定の機能を果たしてきている。

しかし、問題があった。その出力は CSV（カンマ区切り）であるのだが、出力されるファイル形式は、我々が表計算で見慣れた形式ではなかったのである（図１を参照）。そのため、この TECMIN がもつ機能は、十分に活用されないという残念な状態にあった。

では、このフォーマット（一行が「一つの問とそれへの回答」という形式）は特異なものなのであろうか。

本稿では、このフォーマットへの驚きを契機として、この形式は、データ処理の前処理段階では非常に重要な役割を果たすものであることを述べる。そして、その形式を活用する能力は、Google Forms や MicroSoft Forms が出力する結果を活用する上でも有効かつ重要なものであることを述べる。

1.2　TECMIN のアンケートが吐き出す CSV ファイル

では、最初に TECMIN で採取した「アンケート」の結果の CSV ファイルを具体的にみてみたい（図１）。これが TECMIN が出力する CSV の形式である。しかし、我々におなじみの CSV の形式は、表計算の形式がそのままカンマ区切りで出力されているものである（図２）。つまり、アンケートデータの場合、行が個体（回答者）で列に設問、回答カテゴリが並ぶフォーマットである。多くの人は集計作業を Excel で行う。そのために、データを Excel の表計算の形式であつかうことが当たり前になっている。そのために、このロング・フォーマット[2] に直面した時に、どのように処理すればいいのか戸惑うことになる。

	A	B	C	D	E	F
1	回答者	回答日	回数	質問番号	質問内容	回答
2	221 学生1	2018/4/9 14:04	1	1	あなたは、自宅にパソコンを持っていますか？	家族共用で持っている
3	221 学生1	2018/4/9 14:04	1	2	今までにパソコンを使用したことがありますか？	多少ある
4	221	2018/4/9 14:04	1	3	質問２で「ある」または「多少ある」と答えた人は、どのような使い方でしたか？（複数回答可）	ホームページ閲覧
5	221	2018/4/9 14:04	1	4	あなたまたはあなたの家庭は、インターネットに接続していますか？	はい
6	221	2018/4/9 14:04	1	5	質問４で「はい」と答えた人のインターネット接続は、常時接続（定額制）ですか？	わからない
7	221	2018/4/9 14:04	1	6	あなた個人で電子メールアドレス（携帯電話やスマホ以外で）を持っていますか？	いいえ
8	221	2018/4/9 14:04	1	7	あなたは、携帯電話やスマートフォンを持っていますか？	iPhoneを持っている
9	221	2018/4/9 14:04	1	8	あなたは、携帯電話やスマートフォンからインターネット接続や電子メールを利用していますか？	はい
10	221	2018/4/9 14:04	1	9	あなたは、高校時代に作新学院大学のホームページを見たことがありますか？	はい
11	221	2018/4/9 14:04	1	10	あなたの出身高校は、次のうちどれに該当しますか？	普通科
12	221	2018/4/9 14:04	1	11	質問１０で「その他」と答えた人は何科でしたか？具体的に記入してください。（文字入力）	
13	221	2018/4/9 14:04	1	12	あなたが高校で学習した情報科目は、次のどれですか？（複数回答可）	社会と情報
14	221	2018/4/9 14:04	1	13	質問１２で「その他」と答えた人は具体的に記入してください。（文字入力）	
15	221	2018/4/9 14:04	1	14	あなたは、高校の授業でどのようなソフトウェアを利用したことがありますか？（複数回答可）	ワープロ ,表計算
16	221	2018/4/9 14:04	1	15	質問１４で「プログラミング言語」を選択した人は、具体的に記入してください。	
17	221	2018/4/9 14:04	1	16	質問１４で「その他」を選択した人は、具体的に記入してください。	
18	221	2018/4/9 14:04	1	17	あなたが、ワープロや情報処理で合格した検定試験や取得した資格があれば記入してください。	
19	221	2018/4/9 14:04	1	18	使ったことのあるまたは使っているソーシャルメディアサービスをチェックしてください。（複数回答可）	twitter ,YouTube ,Instagram
20	221 学生1	2018/4/9 14:04	1	19	質問１８で「その他」を選択した人は、具体的に記入してください。	
21	221 学生2	2018/4/9 14:05	1	1	あなたは、自宅にパソコンを持っていますか？	家族共用で持っている
22	221	2018/4/9 14:05	1	2	今までにパソコンを使用したことがありますか？	ある
23	221	2018/4/9 14:05	1	3	質問２で「ある」または「多少ある」と答えた人は、どのような使い方でしたか？（複数回答可）	ワープロ ,表計算
24	221	2018/4/9 14:05	1	4	あなたまたはあなたの家庭は、インターネットに接続していますか？	はい
25	221	2018/4/9 14:05	1	5	質問４で「はい」と答えた人のインターネット接続は、常時接続（定額制）ですか？	わからない
26	221 学生2	2018/4/9 14:05	1	6	あなた個人で電子メールアドレス（携帯電話やスマホ以外で）を持っていますか？	いいえ

図１　コンピュター利用アンケートの回答：最初の部分

さて、こうしたデータ表示の仕方（図１）が、冒頭に述べたロング・フォーマットである。これを、R を用いて処理すると、図２のような見慣れた表計算形式（ワイド・フォーマット）になる。しかし、変数（ここでは、19個）の数がおおいため、すべてを表示することはできない（列は T までである。また MA フィールドは未展開である）。後ほど、この問題についてさらに検討を加える。

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	回答者	1	2	3	4	5	6	7	8	9	10	11	12	13	14
2	22 学生1	自分用を持っ	ある	ワープロ ,ホー	はい	いいえ	はい	iPhoneを持っ	はい	いいえ	その他	外国語科	その他	NA	ワープロ ,表計
3	22 学生2	家族共用で持	多少ある	情報発信（ホ	はい	わからない	はい	iPhoneを持っ	はい	はい	その他	総合学科	情報処理	NA	ワープロ ,表計
4	22 学生3	家族共用で持	ある	ホームページ	はい	はい	いいえ	iPhoneを持っ	はい	はい	工業科	NA	情報処理 ,ビジ	NA	利用したこと
5	22	家族共用で持	ある	ワープロ ,ゲー	はい	はい	いいえ	Android系ス	はい	はい	普通科	NA	社会と情報	NA	ワープロ
6	22	自分用を持っ	ある	ホームページ	はい	はい	はい	iPhoneを持っ	はい	はい	普通科	NA	その他	情報	ワープロ ,表計
7	22	家族共用で持	多少ある	ホームページ	はい	わからない	はい	iPhoneを持っ	はい	はい	商業科	NA	情報処理 ,ビジ	NA	ワープロ ,表計
8	22	自分用を持っ	ある	ワープロ ,ホー	はい	はい	はい	Android系ス	はい	はい	普通科	NA	社会と情報	NA	ワープロ
9	22	家族共用で持	ある	ゲーム ,その他	はい	わからない	いいえ	iPhoneを持っ	はい	はい	普通科	NA	社会と情報 ,そ	NA	ワープロ ,表計
10	22	持っていない	多少ある	ホームページ	はい	はい	はい	iPhoneを持っ	はい	はい	普通科	NA	社会と情報 ,ビ	NA	ワープロ ,表計
11	22	家族共用で持	多少ある	ホームページ	はい	はい	はい	iPhoneを持っ	はい	はい	普通科	NA	情報A ,その他	情報	ワープロ ,表計
12	22	自分用を持っ	ある	ワープロ ,ホー	はい	わからない	いいえ	iPhoneを持っ	はい	はい	商業科	NA	ビジネス情報	NA	ワープロ ,ホー
13	22	家族共用で持	ある	その他	はい	わからない	いいえ	iPhoneを持っ	はい	はい	普通科	NA	情報技術基礎	NA	ワープロ
14	22	自分用を持っ	ある	ホームページ	はい	はい	はい	iPhoneを持っ	はい	はい	普通科	NA	社会と情報	NA	表計算 ,ホー
15	22	自分用を持っ	ある	ワープロ ,表計	はい	はい	はい	iPhoneを持っ	はい	いいえ	普通科	NA	社会と情報	NA	ワープロ
16	22	自分用を持っ	ある	ホームページ	はい	はい	はい	iPhoneを持っ	はい	はい	普通科	NA	社会と情報	NA	ワープロ
17	22	自分用を持っ	ある	ワープロ ,表計	はい	はい	はい	iPhoneを持っ	はい	はい	普通科	NA	社会と情報	NA	ワープロ
18	22	持っていない	ある	ワープロ	はい	わからない	はい	iPhoneを持っ	いいえ	はい	工業科	NA	その他	電気	ワープロ
19	22	持っていない	多少ある	表計算 ,ホーム	はい	わからない	いいえ	iPhoneを持っ	はい	はい	普通科	NA	社会と情報	NA	表計算 ,ホー
20	22	持っていない	ない	NA	いいえ	NA	いいえ	iPhoneを持っ	はい	いいえ	普通科	NA	社会と情報	NA	プレゼンテー
21	22	持っていない	多少ある	ホームページ	はい	わからない	いいえ	iPhoneを持っ	はい	はい	普通科	NA	社会と情報	NA	ワープロ ,プレ
22	22	家族共用で持	多少ある	ホームページ	はい	わからない	いいえ	iPhoneを持っ	はい	はい	普通科	NA	社会と情報	NA	ワープロ ,表計
23	22	自分用を持っ	ある	ワープロ ,ホー	はい	わからない	いいえ	Android系ス	はい	いいえ	普通科	NA	情報の科学	NA	ワープロ ,ホー
24	22 学生23	持っていない	ある	ワープロ ,ホー	はい	はい	はい	Android系ス	はい	いいえ	普通科	NA	社会と情報 ,そ	NA	ワープロ ,ホー
25	22 学生24	家族共用で持	多少ある	ホームページ	はい	はい	いいえ	iPhoneを持っ	はい	はい	普通科	NA	社会と情報	NA	ワープロ
26	22	家族共用で持	多少ある	ワープロ ,ホー	はい	わからない	はい	iPhoneを持っ	いいえ	はい	普通科	NA	社会と情報	NA	ワープロ

図２　図１のロング・フォーマットをワイド・フォーマットに変換したもの

再度確認すると、フォーマットの違いは以下のようになっている

ロング・フォーマット（縦長）　　一行「一問 / 回答」
ワイド・フォーマット（横長）　　一行「回答者一人」に関するすべての設問の回答

今、ここで確認しておきたいのは、どちらのフォーマットで表現してもそれが体現している「情報」は等価である、ということである。それを踏まえて問題はそれぞれのフォーマットがどのような処理に適しているのか、ということを検討していく。

2　模式図的なサンプルデータを用いてロング / ワイド変換を整理する

単純なサンプル・データを用いて、ロング・フォーマットに対して、どのような操作をすればワイド・フォーマットにできるか考えることで、ロング・フォーマットとワイド・フォーマットの関係をみていきたい。

2.1　模擬データを用意して、それのロング化、ワイド化の処理を確認する。

1）変数として、「回答者」「性別」「質問１」「質問２」「質問３」をもったデータを考える。回答者には、名前01～名前20を割り当て、性別は、「男性」「女性」を乱数で割り当てる。「質問１」の回答としてはA、B、C、Dを、「質問２」には、a、b、c、d、eを。「質問３」には、1、2、3、4、5をそれぞれ乱数で割り当てて検討に用いるデータを作成した。

図３にあるのが、こうして生成したワイド・フォーマットのサンプルデータである。これを gather() コマンド[3] でロング・フォーマットにしたものが図４である。対応関係をよく見ていただきたい。

	A	B	C	D	E
1	回答者	性別	質問1	質問2	質問3
2	名前01	男性	C	a	3
3	名前02	男性	C	d	4
4	名前03	男性	C	a	5
5	名前04	女性	B	a	5
6	名前05	男性	C	e	3
7	名前06	女性	B	c	1
8	名前07	女性	B	b	2
9	名前08	女性	B	b	5
10	名前09	男性	C	a	5
11	名前10	男性	A	c	4
12	名前11	女性	D	d	5
13	名前12	女性	B	a	2
14	名前13	女性	B	c	1

図３　ワイド・フォーマット

	A	B	C	D
1	回答者	性別	質問	回答
2	名前01	男性	質問1	C
3	名前01	男性	質問2	a
4	名前01	男性	質問3	3
5	名前02	男性	質問1	C
6	名前02	男性	質問2	d
7	名前02	男性	質問3	4
8	名前03	男性	質問1	C
9	名前03	男性	質問2	a
10	名前03	男性	質問3	5
11	名前04	女性	質問1	B
12	名前04	女性	質問2	a
13	名前04	女性	質問3	5

図４　ロング・フォーマット

2.2　図１のロング・フォーマットを図２のワイド・フォーマットに変換してみる

上で確認した方法で、実際のデータ（図１）をワイドに変換すること試みる。この機能は、R[4] のパッケージで提供されている。それを用いて実際に変換してみる[5]。

```
# TECMIN が吐き出した CSV を読み込む
```{r}
fname = "2018 年度新入生コンピュータ経験アンケート_20180414.csv"
.d <- read_csv(file = fname, locale = locale(encoding = "cp932"))
```


long format を wideformat に変換して Excel 用 CSV で書き出す
```{r}
.d %>% select(-回答日,-回数,-質問内容) %>%
  spread(key=質問番号,value=回答) %>%
write_excel_csv("sample2018.csv")
```
```

図 3　変換スクリプト：rmarkdown で記述。

まず最初に、TECMIN が出力した CSV ファイルを read_csv で読み込んでいる。その時、この CSV ファイルは Excel で読み込むことを前提に文字コードが shift-jis(cp932) にエンコードされているので、それを明示する。明示しないと文字化けする。

次に、spread() コマンドを使ってロング・フォーマットをワイド・フォーマットに変換し、それを、write_excel_csv(" ファイル名 ") で、書き出している。この write_excel_csv も default では文字コードを UTF－8でエンコーディングするが、BOM 付きの UTF－8であるため、Excel でも文字化けせずに読み込むことができる。なお、こうやって見慣れたワイド・フォーマットにしたからといって、処理が終わるわけではない。分析作業は、これからである。

ここで問題にしたいのは、ロング・フォーマットとワイド・フォーマットのどちらが「分析」に適しているのか、ということではない。この両者を駆使した処理が必要なのである。列数がすくない場合は、ワイド・フォーマットでも、一覧性は維持される。しかし、全体を見渡せることと、分析しやすいかどうかは別の問題である。Excel で処理するとなると、かなり複雑な処理が必要になったり、部分表を作成してそれを集計する必要に直面したことはないだろうか。これは「ワイド・フォーマットの壁に直面している」のであってそういう時は、ロング・フォーマットに変換するとシンプルな処理で解決できる。そのことは、このロング－ワイド変換は、最初からワイド・フォーマットで出力してくる、Google Foms や MSForms の分析においても有用であることを後に説明する。

なお、ここで用いた spread() というコマンドは、gather() というコマンドと対になっている。ここで整理すると以下のような関係がある。

`spread()`　ロング・フォーマットをワイド・フォーマット（表計算形式）に変換する
`gather()`　ワイド・フォーマットをロング・フォーマットに変換する

ここでやっていることはこれだけである。

このようにワイド・フォーマットにすれば、Excel で展開できて、分析もできる、という目処がたち、一安心、ということになるのであろうか。

ではなぜ、ロングとワイドを相互に変換できるコマンドが提供されているのだろうか。実は、本稿で明らかにしたいポイントはそこにある。TECMI が出力するロング・フォーマットが「特異」でそれを見慣れたものに変換できますよ、ということを示すことが本稿の目的なのではない。

3　tidy なデータというアイデア

筆者の一人藤本が、この `spread/gather` というコマンドのペアに出会ったのは、`tidyverse` というパッケージ群の開発を精力的に続けている Hadely Wickham のコマンド群を使う過程であった。

大規模な調査データを分析する際には、分析対象にしている変数に対する統計処理を行う前に、相当丁寧な前処理が必要になる。それらは、無効処理であり、外れ値の除去であり、必要に応じてウエイトをかける、などである。

Hadley Wickham のコマンド群は、このロング / ワイド変換の他に、`dplyr` と呼ばれるコマンド群、また、UNIX などでは大前提になるコマンドのパイプ処理などからなっている。Hadley は、これらのコマンド群を `tidyverse` つまり `tidy` な宇宙と呼んでいる。`tidyverse` 全体を紹介することは本稿の範囲を超えるが、一言でいえば、処理が可視化され、（その結果）再利用が可能なスクリプト（プログラム）の開発を可能にする処理を実現するというアイデアであり、そのための環境の提供である（Wickham.H 2017:2017）。

こうしたスクリプトを生み出すコマンド群が、`dplyr` を中心にした `tidyverse` だとすると、処理するデータに求められる「正しい」形態が、`tidy` データなのである（西原 2017a, 2017b）。

3.1　tidy データ、整理データ、整然データ

`tidy` データとは、「整理データ」[6] とも「整頓データ」[7] とも訳されているが、ポイントは、問題なく分析処理を遂行できるシンプルなデータ形式という意味である。

「分析準備のためのデータクリーニングに対して膨大な労力が費やされているが、でき

るだけ簡単で効果的にデータクリーニングを行う手法についてはほとんど研究がなされていない。 本論文では、データクリーニングにおいて、小さなことではあるが重要な要素であるデータの整然化に取り組む。整然データセットは、操作・モデル化・視覚化が容易であり、特有の構造を持っている。すなわち、個々の変数が列となり、個々の観測が行となり、個々の観測の構成単位の類型が表となる。 このフレームワークは、幅広い範囲の非整然データセットに対処するためにわずかな数のツールしか必要としないことから、雑然データセットを整然化することを容易にする。 この構造は、データ分析のために、整然ツール、すなわち入力と出力がいずれも整然データセットとなるツールを開発することをも容易にする。 一貫したデータ構造とそれに合ったツールの利点は、面白みのないデータ操作の雑用から解放された事例研究で実証される。」(「整然データ」前書き（Wickham.H 2014:2017)）

3.2　データサイエンスにおける前処理での基本操作

Tokyo.R のコーディネータでもある三村喬生は、R によるデータ処理の流れを説明する際に、パイプ処理で右から左に流れるように記述することができる、という説明に加え、よく次のような表現する。

必要に応じで、データを「縦にしたり」「横にしたり」して処理をしていく

つまり、必要な前処理にとって、適切な形にデータを変形させてパイプ処理でつないでいく。これがデータ分析の基本的フットワークである。こうすることで、どこでなにをやっているのかが、スクリプトとして記録され、そのことが、処理の透明化につながり、再利用なスクリプトを実現するのである[8)]。

3.3　この操作が可能にする柔軟なデータ処理

冒頭のべたように、TECMIN の出力する CSV の形式は、馴染みがないものであった。最近では、Web アンケートを取る場合には、Google Forms や MicroSoft Forms が活用される場面が多くなっている。そして、これらは、表計算型の、つまり、ワイド・フォーマットでデータを出力する。また、簡易集計機能も提供されているから、非常に手軽である。では、これでなにも問題がないのだろうか。

小規模な、つまり設問数が少ないアンケートであれば、表計算フォームで出力されれば、全体も見渡せるし、それでいいのかもしれない。

しかし、設問数が非常に多数あるアンケートになるどうなるか。すでに多くの人が経験していると思うが、文字通り、横に広い、広大に広いフォーマットでの出力が得られてし

まう。コンピュターの画面は、昔で言えば80桁しかなかったのである。グラフィカルなディスプレイの今日、この80という文字数（日本語だと40文字）の数値としての意味は後退しているものの、一画面に収まらないものの把握は困難である。一瞥が困難なだけでなく、処理にあたって、一覧できない対象を操作するのは誤操作のもとになる。また各変数が列に分散していることに伴ない必要な分析のためのデータ整理が複雑になる場合もある。そうした時には、ワイド・フォーマットからロング・フォーマットにすることで、シンプルな処理で対応できるようになる。詳細は、Wickham.H・Grolemud.G2017:2017を参照。tidyverse というパッケージはこのためにある。

こうして、ワイド・フォーマットをロング・フォーマットにする処理が意味をもってくる。

4　実例スクリプト

実例を用いて、ワイド / ロング変換が可能にする処理をご紹介する。

4.1　出席アンケート

2020年は、コロナ 事態に遭遇したため、作新学院大学でも多くの講義がリモート講義となった。そのリモート講義で問題になることの一つが出席記録をどうするかということである。作新学院大学では、冒頭から触れている TECMIN のアンケート機能を用いて、「出席アンケート」に回答させている。この機能は、残念ながら、アンケート間の串刺し集計、つまりある回答者が、いつの講義に出席と回答したかという集計は自動的には行われず、データをダウンロードしたのちに、Excel などで「手動処理」するしかない。

アンケートの質問項目は、以下のようになっている。

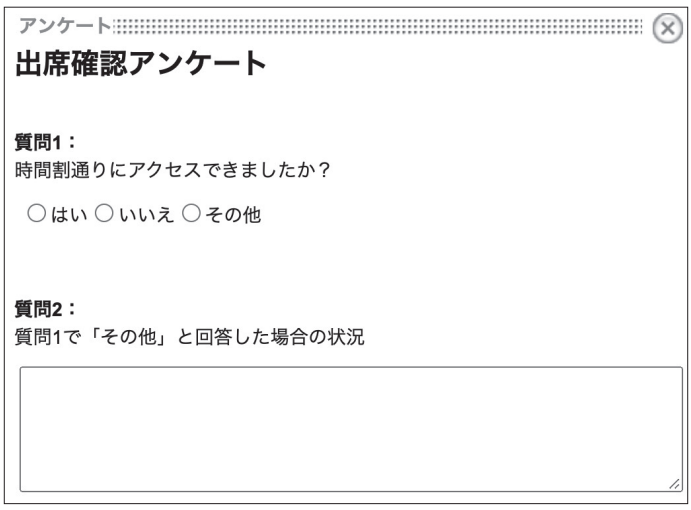


図５　出席アンケートの入力画面

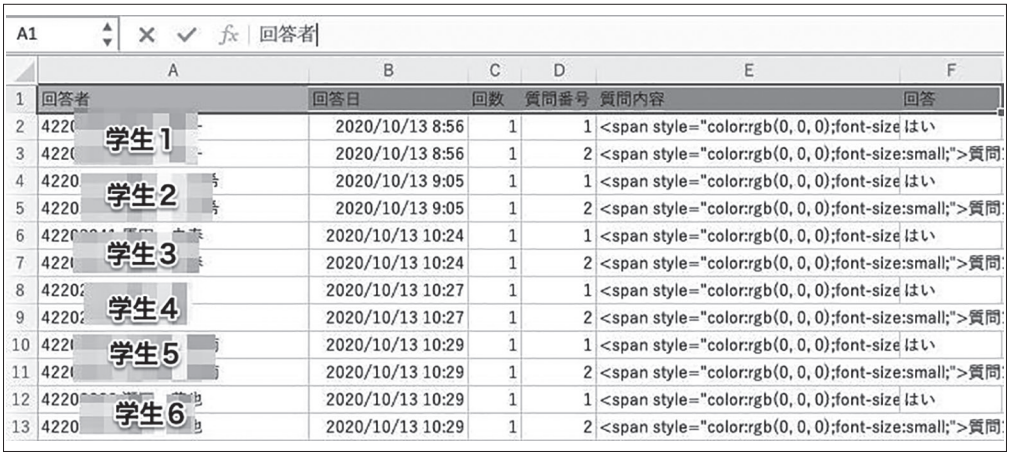
設問数は２つ。それぞれ回答選択肢は、質問１が3、質問２が自由記述、という構成である。そのために、ロング・フォーマットでの出力でも、質問が二つだけなので、回答者一人に対して２行が使われるだけである。これならExcelを使った手作業でもなんとかなる範囲かもしれない。（最初にご覧いただいた「コンピュター経験アンケート」では、質問は19あるので、一人あたり19行が使われてるデータと比べて欲しい。）

これを各回ごとに処理するだけであれば、処理はシンプルである。問題は、第１回から第15回分を学生名ごとに列を統合する処理である。

すべての学生が、毎回全員出席しているなら学籍番号順にソートして、列で連結すればよいことになる。しかし、現実はそうはいかない。講義回ごとに、欠席する学生はまちまちである。

そうしたデータを連結する時に、Excelの手作業で行われていることは、おそらく以下のような作業であろう。

1）履修学生の全員リストを用意する。これは学籍番号でソートされている。
2）連結したい回のデータを学籍番号順にソートして、それを、受講生リストのその回の列にペーストする。
3）学籍番号なり名前が一致していることを目で追いながら、ずれているところがあれば、ずらしカット＆ペーストを繰り返しながら、最後まで見ていく。
4）これを１回目から15回目まで行うことになる。



	A	B	C	D	E	F
1	回答者	回答日	回数	質問番号	質問内容	回答
2	4220	2020/10/13 8:56	1	1	<span style="color:rgb(0, 0, 0);font-size	はい
3	4220	2020/10/13 8:56	1	2	質問	
4	4220	2020/10/13 9:05	1	1	<span style="color:rgb(0, 0, 0);font-size	はい
5	4220	2020/10/13 9:05	1	2	質問	
6	4220	2020/10/13 10:24	1	1	<span style="color:rgb(0, 0, 0);font-size	はい
7	4220	2020/10/13 10:24	1	2	質問	
8	4220	2020/10/13 10:27	1	1	<span style="color:rgb(0, 0, 0);font-size	はい
9	4220	2020/10/13 10:27	1	2	質問	
10	4220	2020/10/13 10:29	1	1	<span style="color:rgb(0, 0, 0);font-size	はい
11	4220	2020/10/13 10:29	1	2	質問	
12	4220	2020/10/13 10:29	1	1	<span style="color:rgb(0, 0, 0);font-size	はい
13	4220	2020/10/13 10:29	1	2	質問	

図6　1人２行のCSVファイル。なお質問内容は、HTMLのタグで挟まれているので除去処理が必要。

では、ここで、こうしたフォーマットで入手したファイルをどう処理するかみていこう。実際の処理スクリプトは、付表にRmdから生成したHTMLとして提示してある。

4.2　処理の手順は、以下のようになる。

1）講義回ごとの「出席アンケート」を読み込む。

2）回答状況を確認する。問２が未記入の場合は NA になっているので、それを確認。それを踏まえて、質問番号１の回答を filter で抽出する。

3）積み重ねるための前処理を行う。

（a）「質問内容」の html タグを除去する。

（b）「回答者」を「学籍番号」と「氏名」に分解し、学籍番号を数値（numeric）に変換する。それらを tmp4～ tmp6として保存

（c）回という列を mutate し、そこに、”第４週”～”第６週”と入れる。

4）bind_rows() でつみかさね、それを spread() で回を key に値を列に展開する。

ここで見るように、単純な処理の積み重ねで、データの結合が可能になる。そののち、必要に応じて、ワイド・フォーマットに展開すればよい。その時、存在しない行要素（回答者）の部分は、デフォルトでは NA となるが、これは spread コマンドの中で、fill=”xxxx” のように置き換えを指定できる。ここでは、" 欠席 " で fill している。

そうして得られたものが以下のワイド・フォーマットである。

```
bind_rows(d4,d5,d6) %>% spread(key=回,value=回答, fill="欠席") %>% head(10) %>% knitr::kable()
```

学籍番号	氏名	第4週	第5週	第6週
42[redacted]	学生１	はい	はい	はい
42[redacted]	学生２	はい	はい	いいえ
42[redacted]	：	はい	はい	はい
42[redacted]	：	はい	はい	はい
42[redacted]	：	はい	はい	はい
42[redacted]	：	はい	はい	はい
42[redacted]	：	はい	はい	はい
42[redacted]	：	はい	はい	はい
42[redacted]	：	はい	はい	はい
42[redacted]	学生10	はい	はい	はい

図７　複数のデータを行の一致を気にすることなく、結合し集計した結果をワイド・フォーマットに変形した

5　まとめ

以上の検討を通して明らかになったのは、以下のことである。

1．ワイド・フォーマットもロング・フォーマットも体現している情報に違いはない。

2．ワイド・フォーマット（いわゆる表計算型）は、最終（報告）形態としては意味を持つこともあるが、分析過程ではこの形式が必ずしも「有効」なわけではない。時に、複雑な処理を分析者に強いることがある。

3．データの分析過程では、ワイド・フォーマットに拘泥することなく、適用する処理に必要な形態に展開すべきである。ワイド⇄ロング変換を適宜適用し処理にとってシンプルな形態を実現すべきである。

4．データのマージ（連結）に関しては、ロング・フォーマットが優れている。また、場合によっては、`dplyr::***_join` を使うことが適していることもある。

注

1）TECMIN 作新学院大学で2012年より使用している学内情報サービス。本体は、国立情報学研究所が開発した NetCommons 2.0 で、本学で使用しているものは、それをベースにベンダーが追加開発を行った、NextCommons である。なお、TECMIN（テクミン）とは、大学のマスコット・キャラクタであり、20xx 年に、当時幼児教育科の赤羽薫教授によってデザインされた。http://..... 。学内情報サービスを TECMIN と呼ぶのは、このマスコット・キャラクタと Toral Education & CoMmunication INfrastructure（教育 / コミュニケーション総合基盤）をかけている。2012年頃に、情報センター委員会で提案された。

なお、従来は外部の業者に委託していた「授業評価アンケート」も TECMIN のアンケート機能を使って実施されている（短大は MS Forms での実施）。

2）ロング・フォーマット、ワイド・フォーマット

これは、データ一覧の形式を示している。我々が Excel のような表計算で参照するのは、後者、ワイド（横長）フォーマットである。このフォーマットでは、一般に、行は個体（回答者）、列は（通常複数、多数の）変数から構成されている。これに対して、ロング・フォーマットというのは、データが保持している情報は、ワイド・フォーマットと同一であるが、行が個体ではなく、設問項目名一つとそれへの回答一つから構成されたものである。

この形式は「tidy データ」と呼ばれ、「整頓データ」や「整理データ」と翻訳されている。本格的な統計分析の前のデータ整理（いわゆる前処理）の段階では、この整頓データを用いることが作業の安定性、再現性、そして結果としての効率化を実現する。Wickham,H 2017:2017参照。

3）パッケージ `tdiyr` で、`gather()`、`spread()` として提供されてきたが、2019年に、`pivot_longer()`、`pivot_wider()` として改定されている。本稿では、それ以前に作成されたスクリプトを用いているので、`gather()`、`spread()` のまま掲載している。

4）プログラミング言語 R、RStudio

本稿で用いるデータ処理言語は、R と RStudio である。R は、統計処理に特化したプログラミング言語で、その出自は、米国ベル研究所で開発された S にある。S を開発した JM チェンバースらは、同じくベル研の Tukey の EDA（探索的データ解析）の思想を共有しており、S そして後継となる R は、データの可視化を重視した実装となっている。ただ、基本的な操作がコマンドラインからのものであるため、GUI 操作に慣れた人からは利用するためのハードルが高いと言われることもあるが、その R に GUI 環境を提供する RCommander（John Fox 教授）も開発され、

一般的な GUI 操作で利用する環境は整備されている。加えて、コマンド・ラインであっても、分析中の関連ファイルを管理するプロジェクトという概念をもち、また、GitHub のようなバージョン管理も組み込んだ、統合開発環境としての RStudio が公開されるにおよび、利用における利便性は格段に向上している。

なお、Excel、Excel＋HAD、そして SPSS のような一般的に統計処理ソフトとの違いは、この GUI かコマンドライン（CUI）かではなく、探索的データ解析（EDA）的なアプローチを実現しやすいところにある。

５）調査票は、付表１を参照されたい。

６）Hadly Wickham『R でデータサイエンス』オライリージャパン

７）「なぜ tidy データを整然データと訳したのか」https://id.fnshr.info/2017/01/09/tidy-data-seizen/

８）スクリプトで処理過程を記述することによって、その処理が再現可能になる。RStudio で用いられる Rmd（Rmarkdown）ファイルは、コマンド処理に加えて、グラフ、分析内容などの記述も再現可能にする。つまり、レポートの生成を「再現可能」にする。詳細は、高橋2018などを参照。

参考文献

Healy, Kieran, 2018, Data Visualization: A Practical Introduction, Princeton Univ Pr,（訳：瓜生真也 , 江口哲史 , 三村喬生 , 2021,『データ可視化入門』講談社）

西原史暁 , 2017a,「整然データとはなにか」https://id.fnshr.info/2017/01/09/tidy-data-intro/

西原史暁 , 2017b,「なぜ“tidy data”を「整然データ」と訳したのか」https://id.fnshr.info/2017/01/09/tidy-data-seizen/

高橋康介 , 2018,『再現可能性のすすめ :RStudio によるデータ解析とレポート作成』共立出版

Tukey, John Wilder, 1977, Exploratory Data Analysis, Pearson

Wickham, H.（2014）. *Tidy data*. Journal of Statistical Software, 59（10）. doi:10.18637/jss.v059.i10（訳：西原史暁 , 2017,「整然データ」https://id.fnshr.info/2017/01/09/trans-tidy-data/）

Wickham,H. Grolemund,G, 2017, *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data,* O'Rilkey,（訳：黒川利明 , 2017,『R ではじめるデータサイエンス』オライリー・ジャパン）

R Core Team（2020）. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

謝辞

本稿執筆過程で、図書館情報センターの平野課長以下職員のみなさん、また、大学教育センターの高橋秀行教授には大変お世話になりました。記して感謝いたします。

付表

１．2019年度コンピュター利用アンケート調査票

２．サンプルデータによるワイド - ロング変換の実例

３．出席アンケート処理　Rmd ファイル

付表１　コンピュータ利用アンケート調査票

2019年度新入生コンピュータ経験アンケート

（TECMINの「学生ルーム」＞「掲示・案内(全学共通)」＞「学生対象アンケート」で参照）

質問1：あなたは、自宅にパソコンを持っていますか？
自分用を持っている、家族共用で持っている、持っていない

質問2：今までにパソコンを使用したことがありますか？
ある、多少ある、ない

質問3：質問２で「ある」または「多少ある」と答えた人は、どのような使い方でしたか？（複数回答可）
ワープロ、表計算、ホームページ閲覧、電子メール、ゲーム、情報発信（ホームページや各種SNS、Facebook、Twitterなど）、その他

質問4：あなたまたはあなたの家庭は、インターネットに接続していますか？
はい、いいえ

質問5：質問４で「はい」と答えた人のインターネット接続は、常時接続（定額制）ですか？
はい、いいえ、わからない

質問6：あなた個人で電子メールアドレス（携帯電話やスマホ以外で）を持っていますか？
はい、いいえ

質問7：あなたは、携帯電話やスマートフォンを持っていますか？
携帯電話（スマホでなく）を持っている、iPhoneを持っている、Android系スマートフォンを持っている、その他スマートフォンを持っている、持っていない

質問8：あなたは、携帯電話やスマートフォンからインターネット接続や電子メールを利用していますか？
はい、いいえ

質問9：あなたは、高校時代に作新学院大学のホームページを見たことがありますか？
はい、いいえ

質問10：あなたの出身高校は、次のうちどれに該当しますか？
普通科、商業科、工業科、農業科、体育科、その他

質問11：質問１０で「その他」と答えた人は何科でしたか？具体的に記入してください。（文字入力）
[　　]

質問12：あなたが高校で学習した情報科目は、次のどれですか？（複数回答可）
社会と情報、情報の科学、情報A、情報B、情報C、情報処理、ビジネス情報、情報技術基礎、その他

質問13：質問１２で「その他」と答えた人は具体的に記入してください。（文字入力）
[　　]

質問14：あなたは、高校の授業でどのようなソフトウェアを利用したことがありますか？（複数回答可）
ワープロ、表計算、ホームページ閲覧、プレゼンテーション、電子メール、データベース、図形描画、プログラミング言語、その他、利用したことはない

質問15：質問１４で「プログラミング言語」を選択した人は、具体的に記入してください。
[　　]

質問16：質問１４で「その他」を選択した人は、具体的に記入してください。
[　　]

質問17：あなたが、ワープロや情報処理で合格した検定試験や取得した資格があれば記入してください。
[　　]

質問18：使ったことのあるまたは使っているソーシャルメディアサービスをチェックしてください。（複数回答可）
mixi、FaceBook、twitter、各種blog、2ちゃんねる、GREE、Mobage、LINE、Google+、YouTube、ニコニコ動画、Instagram、その他、使っていない。もしくは、よく分からない。

質問19：質問１８で「その他」を選択した人は、具体的に記入してください。
[　　]

付表２　ワイドーロング変換スクリプト

wide-long変換

/Users/kazuo/Dropbox/RStudio/紀要12号long-wide/wide2long.Rmd

fujimoto@sakushin-u.ac.jp (mailto:fujimoto@sakushin-u.ac.jp)

2021/01/09

- 1 Wideフォーマットでデータを用意する
- 2 gather コマンドで Wide to long 変換を行う
  - 2.1 その１ 回答者,性別をそのまま、質問1,2,3 を「質問」に、それぞれの回答内容を、「回答」に
  - 2.2 その２ keyとvalueに束ねたいものを「回答者」だけにする
  - 2.3 gather は、pivot_longerを使えと
- 3 spread コマンドを使って、longフォーマットをwideフォーマット変換する。
- 4 参考文献

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.0 ──
```

```
## ✓ ggplot2 3.3.2     ✓ purrr   0.3.4
## ✓ tibble  3.0.4     ✓ dplyr   1.0.2
## ✓ tidyr   1.1.2     ✓ stringr 1.4.0
## ✓ readr   1.4.0     ✓ forcats 0.5.0
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(stringr)
```

1 Wideフォーマットでデータを用意する

- データ数 20
- 変数
  - 回答者
  - 性別
  - 質問1〜3
    - 回答1〜3

```
name <- str_c("名前",str_pad(1:20, 2, pad=0))
set.seed(123)
sex <- sample(c("男性","女性"),20,replace = TRUE)
回答1 <- c("A","B","C","D")
回答2 <- c("a","b","c","d","e")
回答3 <- c("1":"5")
set.seed(123)
data.frame(回答者=factor(name),性別=factor(sex),質問1= factor(sample(回答1,20,replace = TRUE
)),
           質問2= factor(sample(回答2,20,replace = TRUE)),
           質問3= factor(sample(回答3,20,replace = TRUE))) -> .d
.d
```

```
##    回答者 性別 質問1 質問2 質問3
## 1  名前01 男性    C    a    3
## 2  名前02 男性    C    d    4
## 3  名前03 男性    C    a    5
## 4  名前04 女性    B    a    5
## 5  名前05 男性    C    e    3
## 6  名前06 女性    B    c    1
## 7  名前07 女性    B    b    2
## 8  名前08 女性    B    b    5
## 9  名前09 男性    C    a    5
## 10 名前10 男性    A    c    4
## 11 名前11 女性    D    d    5
## 12 名前12 女性    B    a    2
## 13 名前13 女性    B    c    1
## 14 名前14 男性    A    e    1
## 15 名前15 女性    B    d    3
## 16 名前16 男性    C    b    1
## 17 名前17 女性    D    e    5
## 18 名前18 男性    A    a    1
## 19 名前19 男性    C    a    2
## 20 名前20 男性    C    b    4
```

```
.d %>% summary()
```

```
##     回答者     性別    質問1 質問2 質問3
## 名前01 : 1   女性: 9   A:3   a:7   1:5
## 名前02 : 1   男性:11   B:7   b:4   2:3
## 名前03 : 1             C:8   c:3   3:3
## 名前04 : 1             D:2   d:3   4:3
## 名前05 : 1                   e:3   5:6
## 名前06 : 1
## (Other):14
```

2 gather コマンドで Wide to long 変換を行う

- key　と value には、longにしたときの変数名を与える
- 「-」で指定された列名（ここでは、回答者,性別）は、keyとvalueには束ねられない。

```
.d %>% pivot_longer(col = -c(回答者,性別), names_to = "質問", values_to = "回答") -> .d.pivot_lon
g
.d.pivot_long
```

```
## # A tibble: 60 x 4
##    回答者 性別  質問  回答
##    <fct>  <fct> <chr> <fct>
##  1 名前01 男性  質問1 C
##  2 名前01 男性  質問2 a
##  3 名前01 男性  質問3 3
##  4 名前02 男性  質問1 C
##  5 名前02 男性  質問2 d
##  6 名前02 男性  質問3 4
##  7 名前03 男性  質問1 C
##  8 名前03 男性  質問2 a
##  9 名前03 男性  質問3 5
## 10 名前04 女性  質問1 B
## # … with 50 more rows
```

3 spread コマンドを使って、longフォーマットをwideフォーマット変換する。

```
.d.long %>% spread(key=質問,value=回答)
```

```
##    回答者 性別 質問1 質問2 質問3
##  1 名前01 男性    C    a    3
##  2 名前02 男性    C    d    4
##  3 名前03 男性    C    a    5
##  4 名前04 女性    B    a    5
##  5 名前05 男性    C    e    3
##  6 名前06 女性    B    c    1
##  7 名前07 女性    B    b    2
##  8 名前08 女性    B    b    5
##  9 名前09 男性    C    a    5
## 10 名前10 男性    A    c    4
## 11 名前11 女性    D    d    5
## 12 名前12 女性    B    a    2
## 13 名前13 女性    B    c    1
## 14 名前14 男性    A    e    1
## 15 名前15 女性    B    d    3
## 16 名前16 男性    C    b    1
## 17 名前17 女性    D    e    5
## 18 名前18 男性    A    a    1
## 19 名前19 男性    C    a    2
## 20 名前20 男性    C    b    4
```

4 参考文献

- gather, spread

2.1 その１ 回答者,性別をそのまま、質問1,2,3 を「質問」に、それぞれの回答内容を、「回答」に

```
.d %>% gather(key=質問, value=回答,-回答者,-性別) %>% arrange(回答者) -> .d.long
```

```
## Warning: attributes are not identical across measure variables;
## they will be dropped
```

```
.d.long %>% head()
```

```
##   回答者 性別  質問 回答
## 1 名前01 男性 質問1    C
## 2 名前01 男性 質問2    a
## 3 名前01 男性 質問3    3
## 4 名前02 男性 質問1    C
## 5 名前02 男性 質問2    d
## 6 名前02 男性 質問3    4
```

```
.d.long %>% tail()
```

```
##    回答者 性別  質問 回答
## 55 名前19 男性 質問1    C
## 56 名前19 男性 質問2    a
## 57 名前19 男性 質問3    2
## 58 名前20 男性 質問1    C
## 59 名前20 男性 質問2    b
## 60 名前20 男性 質問3    4
```

2.2 その２ keyとvalueに束ねたいものを「回答者」だけにする

- 性別も「質問」に束ねられることになる。

```
.d %>% gather(key=質問, value=回答, -回答者) %>% arrange(回答者) %>% head()
```

```
## Warning: attributes are not identical across measure variables;
## they will be dropped
```

```
##   回答者  質問 回答
## 1 名前01  性別 男性
## 2 名前01 質問1    C
## 3 名前01 質問2    a
## 4 名前01 質問3    3
## 5 名前02  性別 男性
## 6 名前02 質問1    C
```

2.3 gather は、pivot_longerを使えと

出席アンケート処理

/Users/kazuo/Dropbox/RStudio/紀要12号long-wide/出席アンケート処理.Rmd

fujimoto@sakushin-u.ac.jp (mailto:fujimoto@sakushin-u.ac.jp)

10/26/2020

- 1 データを取得する
  - 1.1 TECMINで表示される「調査票」
  - 1.2 講義回ごとに出席アンケートを読み込む
- 2 TECMINから取得した出席アンケートのデータの構成を確認する
  - 2.1 回答状況を確認する
  - 2.2 積み重ねるための前処理
  - 2.3 各回の出席データを積み重ねる
- 3 spreadで、回を列に展開。


```
knitr::opts_chunk$set(echo = TRUE)
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.0 ──
```

```
## ✓ ggplot2 3.3.2     ✓ purrr   0.3.4
## ✓ tibble  3.0.4     ✓ dplyr   1.0.2
## ✓ tidyr   1.1.2     ✓ stringr 1.4.0
## ✓ readr   1.4.0     ✓ forcats 0.5.0
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

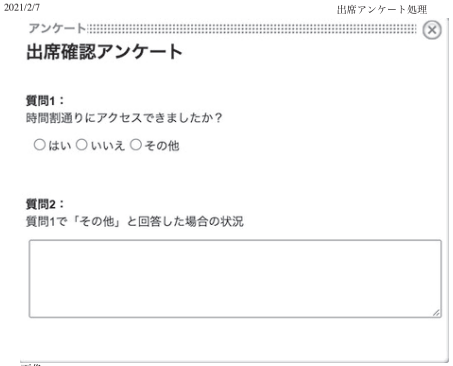
```
library(readxl)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

1 データを取得する

1.1 TECMINで表示される「調査票」



画像

1.2 講義回ごとに出席アンケートを読み込む

```
c_types = cols(
  回答者 = col_character(),
  回答日 = col_datetime(format ="%Y/%m/%d %H:%M:%S"),# 時刻データを評価する
  回数 = col_double(),
  質問番号 = col_double(),
  質問内容 = col_character(),
  回答 = col_character()
)

.d_4 <- read_csv("../../_Dialy2020/作新学院大学/CD2020/出席アンケートCSV/出席確認アンケート 第4週(10_13).csv",col_types = c_types)
.d_5 <- read_csv("../../_Dialy2020/作新学院大学/CD2020/出席アンケートCSV/出席確認アンケート 第5週(10_20).csv",col_types = c_types)
.d_6 <- read_csv("../../_Dialy2020/作新学院大学/CD2020/出席アンケートCSV/出席確認アンケート第6集.csv",col_types = c_types)
```

2 TECMINから取得した出席アンケートのデータの構成を確認する

```
.d_4 %>% head(10) %>% knitr::kable()
```

回答者	回答日	回数	質問番号	質問内容	回答
422[redacted]	2020-10-13 08:56:12	1	1	時間割通りにアクセスできましたか？	はい
422[redacted]	2020-10-13 08:56:12	1	2	質問1で「その他」と回答した場合の状況	NA
422[redacted]	2020-10-13 09:05:00	1	1	時間割通りにアクセスできましたか？	はい
422[redacted]	2020-10-13 09:05:00	1	2	質問1で「その他」と回答した場合の状況	NA

回答者	回答日	回数	質問番号	質問内容	回答
422[redacted]	2020-10-13 10:24:56	1	1	時間割通りにアクセスできましたか？	はい
422[redacted]	2020-10-13 10:24:56	1	2	質問1で「その他」と回答した場合の状況	NA
422[redacted]	2020-10-13 10:27:14	1	1	時間割通りにアクセスできましたか？	はい
422[redacted]	2020-10-13 10:27:14	1	2	質問1で「その他」と回答した場合の状況	NA
422[redacted]	2020-10-13 10:29:38	1	1	時間割通りにアクセスできましたか？	はい
422[redacted]	2020-10-13 10:29:38	1	2	質問1で「その他」と回答した場合の状況	NA

2.1 回答状況を確認する

```
.d_4 %>% filter(is.na(回答))
```

```
## # A tibble: 100 x 6
##   回答者   回答日              回数 質問番号 質問内容          回答
##   <chr>    <dttm>             <dbl>    <dbl> <chr>           <chr>
## 1 422[redacted] 2020-10-13 08:56:12     1        2 "<span style=\"color:rgb… <NA>
## 2 422[redacted] 2020-10-13 09:05:00     1        2 "<span style=\"color:rgb… <NA>
## 3 422[redacted] 2020-10-13 10:24:56     1        2 "<span style=\"color:rgb… <NA>
## 4 422[redacted] 2020-10-13 10:27:14     1        2 "<span style=\"color:rgb… <NA>
## 5 422[redacted] 2020-10-13 10:29:38     1        2 "<span style=\"color:rgb… <NA>
## 6 422[redacted] 2020-10-13 10:29:42     1        2 "<span style=\"color:rgb… <NA>
## 7 422[redacted] 2020-10-13 10:30:07     1        2 "<span style=\"color:rgb… <NA>
## 8 422[redacted] 2020-10-13 10:31:06     1        2 "<span style=\"color:rgb… <NA>
## 9 422[redacted] 2020-10-13 10:31:30     1        2 "<span style=\"color:rgb… <NA>
## 10 42[redacted] 2020-10-13 10:31:40     1        2 "<span style=\"color:rgb… <NA>
## # … with 90 more rows
```

```
.d_4 %>% filter(質問番号==1)
```

```
## # A tibble: 100 x 6
##   回答者   回答日              回数 質問番号 質問内容          回答
##   <chr>    <dttm>             <dbl>    <dbl> <chr>           <chr>
## 1 42[redacted] 2020-10-13 08:56:12     1        1 "<span style=\"color:rgb… はい
## 2 42[redacted] 2020-10-13 09:05:00     1        1 "<span style=\"color:rgb… はい
## 3 42[redacted] 2020-10-13 10:24:56     1        1 "<span style=\"color:rgb… はい
## 4 42[redacted] 2020-10-13 10:27:14     1        1 "<span style=\"color:rgb… はい
## 5 42[redacted] 2020-10-13 10:29:38     1        1 "<span style=\"color:rgb… はい
## 6 42[redacted] 2020-10-13 10:29:42     1        1 "<span style=\"color:rgb… はい
## 7 42[redacted] 2020-10-13 10:30:07     1        1 "<span style=\"color:rgb… はい
## 8 42[redacted] 2020-10-13 10:31:06     1        1 "<span style=\"color:rgb… はい
## 9 42[redacted] 2020-10-13 10:31:30     1        1 "<span style=\"color:rgb… はい
## 10 4[redacted] 2020-10-13 10:31:40     1        1 "<span style=\"color:rgb… はい
## # … with 90 more rows
```

```
.d_4 %>% filter(質問番号==2)
```



```
## # A tibble: 100 x 6
##   回答者   回答日              回数 質問番号 質問内容          回答
##   <chr>    <dttm>             <dbl>    <dbl> <chr>           <chr>
## 1 42[redacted] 2020-10-13 08:56:12     1        2 "<span style=\"color:rgb… <NA>
## 2 42[redacted] 2020-10-13 09:05:00     1        2 "<span style=\"color:rgb… <NA>
## 3 42[redacted] 2020-10-13 10:24:56     1        2 "<span style=\"color:rgb… <NA>
## 4 42[redacted] 2020-10-13 10:27:14     1        2 "<span style=\"color:rgb… <NA>
## 5 42[redacted] 2020-10-13 10:29:38     1        2 "<span style=\"color:rgb… <NA>
## 6 42[redacted] 2020-10-13 10:29:42     1        2 "<span style=\"color:rgb… <NA>
## 7 42[redacted] 2020-10-13 10:30:07     1        2 "<span style=\"color:rgb… <NA>
## 8 42[redacted] 2020-10-13 10:31:06     1        2 "<span style=\"color:rgb… <NA>
## 9 42[redacted] 2020-10-13 10:31:30     1        2 "<span style=\"color:rgb… <NA>
## 10 4[redacted] 2020-10-13 10:31:40     1        2 "<span style=\"color:rgb… <NA>
## # … with 90 more rows
```

```
.d_4 %>% filter(質問番号==3)
```

```
## # A tibble: 0 x 6
## # … with 6 variables: 回答者 <chr>, 回答日 <dttm>, 回数 <dbl>, 質問番号 <dbl>,
## #   質問内容 <chr>, 回答 <chr>
```

2.2 積み重ねるための前処理

- 「質問内容」のhtmlタグを消去する
- 回答者を「学籍番号」と「氏名」に分解し、学籍番号を数値（numeric）に変換する
- それを　tmp* として保存する。

```
.d_4 %>% separate(質問内容,into = c("A","B","C"),sep = ">") %>% select(-A,-C) %>% separate(B,
into=c("B1","B2"),sep="<") %>% select(-B2) %>% filter(!is.na(回答)) %>% separate(回答者,into =
c("学籍番号","氏名"),sep = " ") %>% mutate(学籍番号=parse_integer(学籍番号)) -> tmp4

.d_5 %>% separate(質問内容,into = c("A","B","C"),sep = ">") %>% select(-A,-C) %>% separate(B,
into=c("B1","B2"),sep="<") %>% select(-B2) %>% filter(!is.na(回答)) %>% separate(回答者,into =
c("学籍番号","氏名"),sep = " ") %>% mutate(学籍番号=parse_integer(学籍番号)) -> tmp5

.d_6 %>% separate(質問内容,into = c("A","B","C"),sep = ">") %>% select(-A,-C) %>% separate(B,
into=c("B1","B2"),sep="<") %>% select(-B2) %>% filter(!is.na(回答)) %>% separate(回答者,into =
c("学籍番号","氏名"),sep = " ") %>% mutate(学籍番号=parse_integer(学籍番号)) -> tmp6
```

2.3 各回の出席データを積み重ねる

```
tmp4 %>% select(学籍番号,氏名,回答) %>% mutate(回="第4週") %>% arrange(学籍番号) -> d4
tmp5 %>% select(学籍番号,氏名,回答) %>% mutate(回="第5週") %>% arrange(学籍番号) -> d5
tmp6 %>% select(学籍番号,氏名,回答) %>% mutate(回="第6週") %>% arrange(学籍番号) -> d6
```

3 spreadで、回を列に展開。

- そこでの値を回答に指定。そうすると残りの学籍番号、氏名が行になる。つまり、この方法を使えば、実施「回」ごとに収集したデータを一度、縦積みにして（つまりlongフォーマット）、それを横長に展開することで、おなじみの表が得られる。

2021/2/7 出席アンケート処理

```
bind_rows(d4,d5,d6) %>% spread(key=回,value=回答, fill="欠席") %>% head(10) %>% knitr::kable()
```

学籍番号　氏名	第4週	第5週	第6週
422	はい	はい	はい
422	はい	はい	いいえ
422	はい	はい	はい
422	はい	はい	はい
422	はい	はい	はい
422	はい	はい	はい
422	はい	はい	はい
422	はい	はい	はい
422	はい	はい	はい
422	はい	はい	はい